

REMARKS

Status of Claims

No changes to the claims are made by way of this Amendment. Thus, claims 1-33 are currently pending with claims 1, 6, 9, 14, 19, 24, and 29 being independent.

Office Action

In the August 10, 2005, Office Action, the Examiner rejected claims 1-13 under 35 USC 101 as being directed to non-statutory subject matter, rejected claims 1-3, 6, 9-11, 14-16, 19-21, 24-26, and 29-31 under 35 USC 102(e) as being anticipated by Blakeley (U.S. Patent No. 5,826,077), rejected claims 4, 7, 12, 17, 22, 27, and 32 under 35 USC 103(a) as being unpatentable over Blakeley in view of Steele (U.S. Application Publication No. 2001/0056420), and rejected claims 5, 8, 13, 18, 23, 28, and 33 under 35 USC 103(a) as being unpatentable over Blakeley in view of Agesen (U.S. Patent No. 6,711,672).

The Rejection of Claims 1-13 under 35 USC 101

The Examiner contends that claims 1-13 constitute unpatentable subject matter under 35 USC 101 because they fail to advance the technological arts by being a "mere abstract idea" (page 2). Specifically, the Examiner states that because some of the claimed steps, namely receiving and converting queries, may be performed without a computer, the corresponding claims must be unpatentable subject matter.

Applicant respectfully asserts that claims 1-13 constitute patentable subject matter under controlling precedent and that the Examiner has misconstrued the applicable law. First, the Examiner contends that a method of database management (i.e. claims 1-13) does not advance the technological arts because the method is an "abstract idea." Applicant submits that the Examiner has confused the requirement for producing "a useful, concrete, and tangible result", which requires more than a mere abstract idea, with the requirement that an invention advance the technological arts, which merely requires an invention have some utility. As a database management method

advances the ability to archive information (by converting query language statements into imperative language statements for execution), the present invention clearly promotes the useful arts and sciences.

Additionally, the present invention as defined in claims 1-13 is more than an “abstract idea.” An invention is more than an abstract idea when it produces a “useful, concrete, and tangible result.” *State Street Bank and Trust Co. v. Signature Financial Group.*, 149. F.3d 1368 (Fed. Cir. 1998), also see *MPEP (8th Ed., Rev. 2) § 2106*, page 2100-18, left column. Claims 1-13 produce a tangible result by receiving a query language statement, converting the queries into an imperative language statement, and executing the imperative language statement. Thus, claims 1-13 involve more than merely manipulating “abstract” ideas. For example, the converted query language statements are executed to perform a task (database management) in environments where it would be otherwise impossible or difficult (i.e. a PDA lacking the ability to itself execute pure query language statements). Such functionality is directly analogous to the patentable subject matter defined in *MPEP § 2106* (page 2100-18), including the first listed example of a method for controlling internal data transfer that enables frequently used data to be readily available.

Further, the Examiner appears to apply a different standard under 35 USC 101 to claims 1-13, which include a method, than the remaining claims, which include program products or systems. Such a distinction is inappropriate as “it is of little relevance whether a claim is drawn to a machine or a process ... (t)he legal principals are the same”. *AT&T Corp. v. Excel Communications.*, 172 F.3d 1352 (Fed. Cir. 1999).

The Rejection of Claims 1-33 under 35 USC 102 and 103

The rejections of all pending claims under 35 USC 102 and 103 rely primarily upon the teachings of Blakeley to establish the existence of the claimed subject matter in the prior art. Specifically, the Examiner contends that Blakeley discloses (a) receiving queries in query language, (B) interpreting the queries by associating at declarative function with the query terms, (c) converting the queries to imperative language statements, and (d) executing the imperative language statements.

Applicant respectfully submits that Blakeley's "Object Query Language (OQL)" fails to disclose or suggest various claimed features of the present invention, including receiving queries in a query language and interpreting the queries by associating declarative language functions with the queries.

All independent claims of the present invention generally recite the feature of "receiving queries in a query language." The Examiner contends that columns 8-10 of Blakeley generally disclose this feature based on the functionality of the disclosed Object Query Language (OQL). However, Blakeley, and the disclosed OQL, do not disclose or suggest receiving queries in a query language.

Instead, as is evidenced by the Examiner's own citations, Blakeley provides an extended C++ language (column 7, lines 20-22) that enables a programmer to use query statements (SQL relational calculus) within C++ functions and syntax (column 9, lines 66-67). The query statements utilized by Blakeley are not a query in a query language as recited by the present invention, but instead Blakeley's query statements "borrows the SFW paradigm of SQL" (column 12, lines 20-23) to provide a way "to combine ... C++ with the industry's standard database query language, SQL" (column 19, lines 10-14).

Thus, the Blakeley OQL does not "receive" anything as it merely enables a programmer to include SQL-like statements in C++ code (column 7, lines 1-6). Further, the Blakeley OQL does not receive "queries" in "a query language" as it only allows SQL-like statements to be embedded within a declarative language (column 7, lines 8-22). For example, Blakeley discloses that "the query processor takes as input OQL[C++]..." and not a query in a query language (column 19, lines 27-28). Thus, Blakeley provides no disclosure or suggestion regarding receiving query statements in SQL (or any other query language) as the Blakeley method and system *begins* with creating and receiving OQL[C++] and not with a query language (see FIG. 8).

Additionally, independent claims 1, 14, and 24 each generally recite the feature of "interpreting the queries by associating at least one declarative language function with the query terms." Similarly, claim 6 recites "converting the query language to a plurality of imperative

language statements “ and claims 9, 19, and 29 recite “representing the queries in accordance with a declarative language paradigm.”

In a similar manner as described above, the Examiner contends that Blakeley column 10 through 12, relating to the OQL, disclose these features. However, the Blakeley OQL does not disclose or suggest these features as the OQL does not “receive queries”, as discussed above and the OQL does not “interpret”, “convert”, or “represent” queries through association with declarative language functions.

Specifically, the Blakeley OQL merely enables a programmer to include SQL-like statements within C++ such that the programmer does not “interpret” anything as the SQL-like statements may be directly included within the C++ code (column 8, lines 43-54). In contrast, the claims of the present invention interpret the queries by associating them with a declarative language function.

For example, the present invention may interpret queries without actual knowledge of the query language (paragraph 0075) while the Blakeley OQL requires precise knowledge of the SQL-like query language in order to include the queries within C++ (column 19, lines 10-15). Further, the present invention interprets by providing declarative-language functions that implement various constructs such as filters, joins, sorting, and grouping operators (paragraph 0119). In contrast, the Blakeley OQL merely enables SQL-like statements to be included within C++, such that the SQL-like statements are not interpreted or otherwise transformed (column 9, lines 66-67).

Thus, Blakeley fails to disclose or suggest all claimed features of the present invention, including receiving a query statement and interpreting the query statement utilizing a declarative-language function, as Blakeley’s OQL merely provides an extended version of C++ that allows inclusion of SQL-like statements.

Additionally, the inclusion of Steele and Agesen in the various rejections under 35 USC 103 does not cure the deficiencies of Blakeley. Specifically, Steele and/or Agesen do not disclose or suggest receiving query statements, interpreting, converting, or representing the query statements with declarative language functions, and generally converting the declarative language functions to

an imperative language. Further, the Examiner does not contend that Steele and/or Agesen disclose such features as neither Steele nor Agesen disclose queries of any sort.

Dependent claims recite further claim elements neither taught nor suggested by the art of record. For example, claims 2, 15, and 25 recite “converting the query language to an intermediate tree representation corresponding to the at least one declarative language function associated with the plurality of query terms, and thereafter converting the query to at least one data structure that is interpreted by an imperative language interpreter core to perform the queries,” which is neither taught nor suggested by any reference of record. Claim 10, 20, and 30 recite a similar claim element and are similarly patentable for at least these reasons. The Examiner’s contention that Blakeley (column 13, line 20 to column 14, line 15) discloses this feature is incorrect as the cited portion of Blakeley’s OQL discusses the principals of substitutability and inheritance and not “converting the query language to an intermediate tree representation” as recited by the present invention.

Conclusion

Claims 1-13 are patentable subject matter under 35 USC 101 as they constitute more than a mere abstract idea by providing a “useful, tangible, and concrete result.” Further, the Examiner’s cited references, Blakeley, Steel, and Agesen, alone or in combination, fail to disclose or suggest all claimed features of the present invention. Specifically, Blakeley’s OQL merely provides an extended version of C++ that allows inclusion of SQL-like statements and does not interpret, convert, or otherwise represent query statements received in a query language. Accordingly, Applicants respectfully request that a timely Notice of Allowance be issued in this case.

Should the Examiner have any questions, please contact the undersigned at (800) 445-3460. While the undersigned does not believe any additional fees are due in connection with this application, the Commissioner is hereby authorized to charge any additional fees associated with this communication or credit any overpayment to Deposit Account No. 09-0460.

Application No. 10/090,068
Amendment dated October 21, 2005
Reply to Office Action of August 10, 2005

Respectfully submitted,

HOVEY WILLIAMS LLP

By:

A handwritten signature in dark ink, appearing to read "S. R. Brown", is written over a horizontal line.

Scott R. Brown, Reg No. 40,535
HOVEY WILLIAMS LLP
2405 Grand Boulevard, Suite 400
Kansas City, Missouri 64108
(816) 474-9050

ATTORNEYS FOR APPLICANTS